Final Project
Task : ASR + Translation

Team name : NTU_r06942112_final

Team members :

Student ID	Name	Word division
R06942098	曾柏偉	模型、資料處理、報告
R06942112	張嘉麟	Team sheet
R06943124	王鈺凱	
R06943121	蕭芳宗	differ.lib -> seq match

Data Preprocessing :

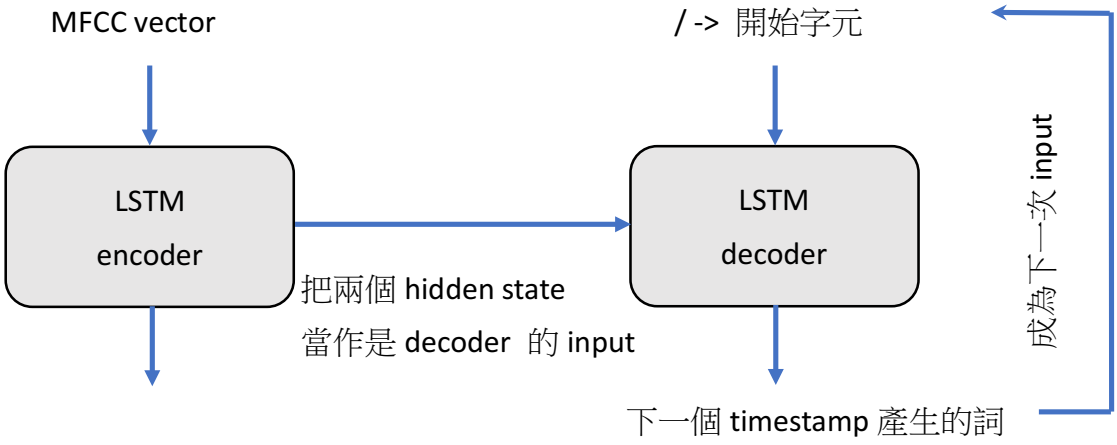
這部分因為 MFCC data 上每一個長短都不一樣，所以為了能夠使用 GPU 去運算我把全部都 pad 0(在每一筆資料尾端)到維數是 (246,39)且共有 45036 筆資料，在 testing data 上面的處理也是照以上的方法去做。至於在每一題選項(caption)上面的處理，我原本使用 pre-trained 的詞向量(word vector)，但是後來發現字詞的如果都是單一的去切的話好像沒有考量到一些中文語意，我後來就決定先使用 jieba 去斷詞，然後再把這些斷好的詞送進 genism word2Vec model 去產生詞向量，但是因為詢問助教的意見之後，我把這些詞向量當作 embedding_layer 的 weights 並且跟著我的模型一起訓練(也就是說 trainable=True)，在測試集上面的處理也是參照上面的方式。

Model Description and Discussion :

1. Teaching_forcing model :

   這個模型是我在過 simple baseline 時後使用的 model，並且參照 keras tutorial 並且加入了摺積層(Conv1D)，出來的 kaggle score = 0.33700 ，在訓練了大概 50 epochs 後，因為這個是 seq2seq based 的我都會先看出來的句子大概是長得怎樣，其實雖然出來的句子是有文

法的，但是比照測試集的選項時，卻發現相似程度是真的滿低的，所以後來我用一個內建的詞意相近的套件(difflib.SequenceMatcher)，去判斷到底哪個選項跟我產生的句子辭意是相近的，以下是我的模型示意圖。

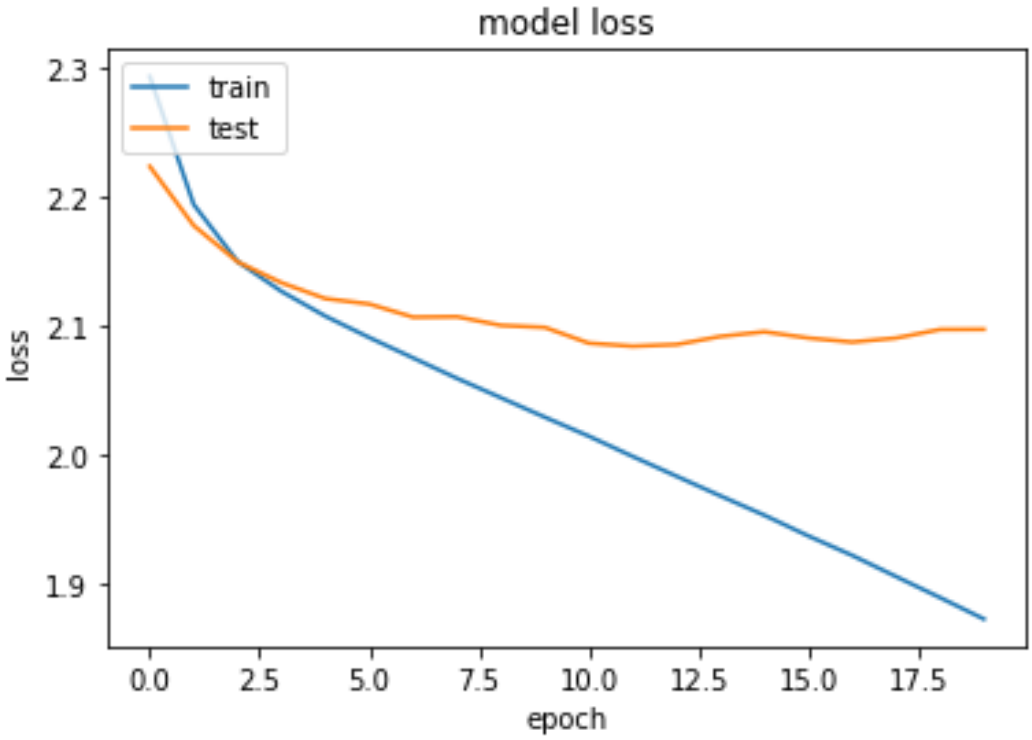


以下是參數個數的圖：

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, None, 39)	0	
input_2 (InputLayer)	(None, None, 300)	0	
lstm_1 (LSTM)	[(None, 300), (None,	408000	input_1[0][0]
lstm_2 (LSTM)	[(None, None, 300), (	721200	input_2[0][0] lstm_1[0][1] lstm_1[0][2]
dense_1 (Dense)	(None, None, 2448)	736848	lstm_2[0][0]

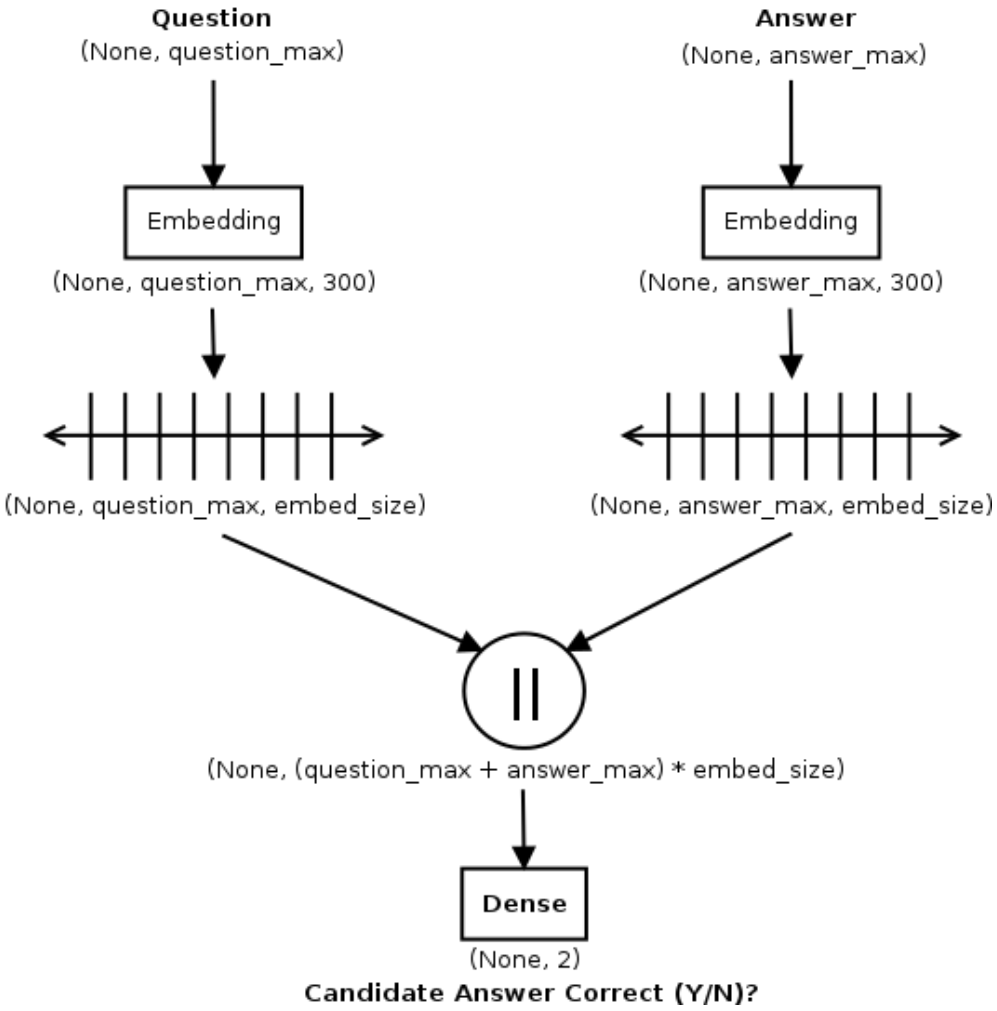
Total params: 1,866,048
Trainable params: 1,866,048
Non-trainable params: 0

以下是我訓練 20 epochs 的 loss 變化程度的示意圖：



2. Retrieval-based (Question-answering)

這個方式雖然我最後沒有訓練起來，不過我有和助教討論過這應該是可行的，這個方法必須要製作對的選項和錯的選項，也就是說，把原本助教給的訓練集資料(training data)設置為 label->1，而我自己在做出兩組 label->0 的錯誤選項。而在測試集時，我就必須對四個選項去預測然後選取 label->1 機率為最大者的那一項。最後因為依照助教的建議改完之後，模型仍然無法收斂，所以最後就沒有再訓練了。

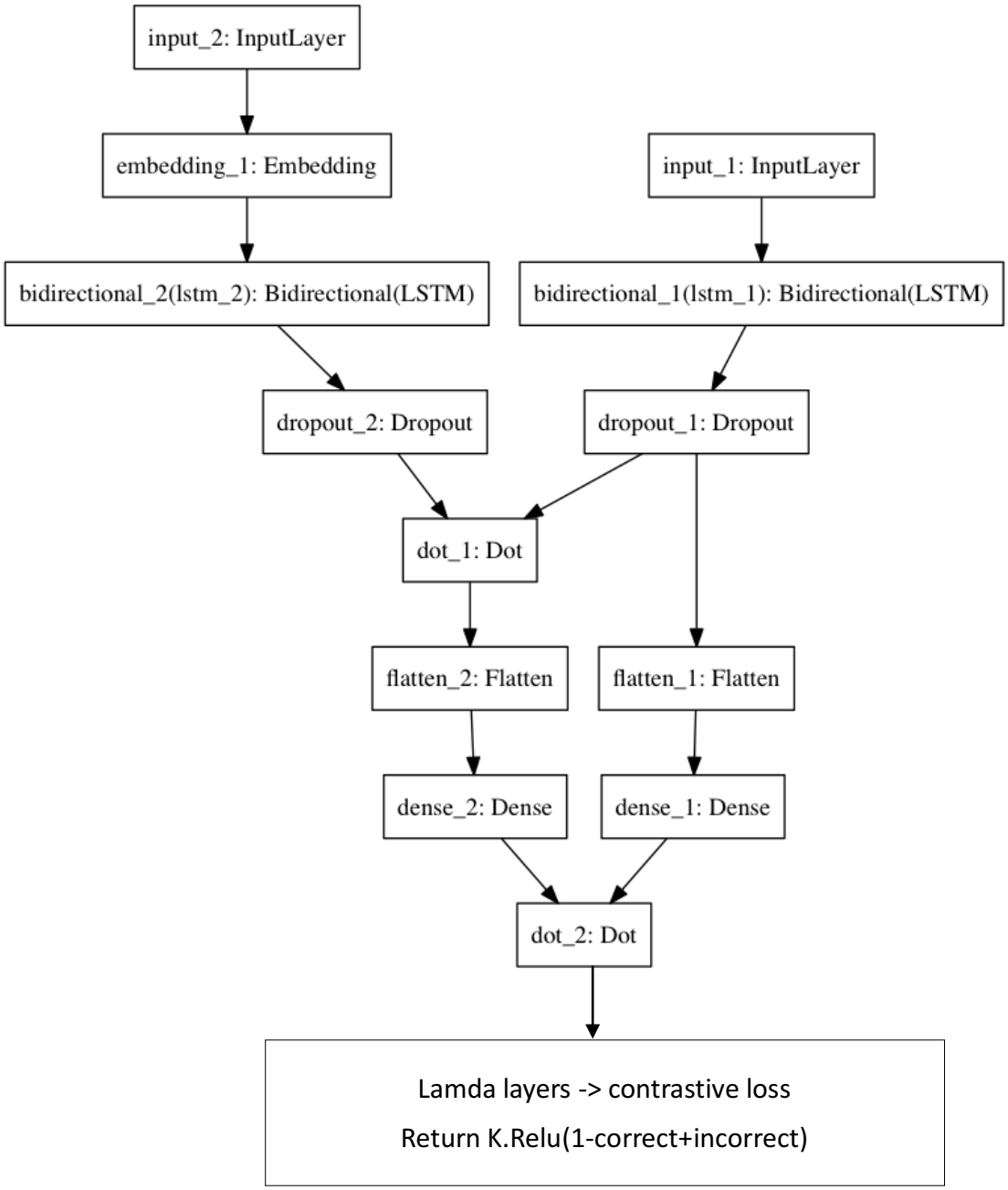


以上是這個 QA 模型的示意圖，經過 embedding layer 後，我只把最後一個 time stamp 留起來(return_sequence=False)，然後把兩個串起來之後送進入一層 dense，在這個 task 雖然不可行，可是我在其他 task 我在試過是可以的！至於 training loss 的圖因為在我這邊的準確率竟然可以達到接近一，所以我就沒有附上了。

3. Retrieval–based (hinge_loss) :

這個模型是我過 strong baseline 的模型，平均一個模型的上傳 kaggle 的分數都在 0.49~0.52 之間，所以最後我使用 ensemble 就通過了 strong baseline，也很感謝助教給我這麼多建議才在死線前完成。使用這個模型必須要輸出是一個 hinge loss 的分數，以下就是我使用 keras (plot_model)畫出來的圖：是 siamese model 架構的感覺，最後 Output 的 Dot 我有 normalization，並且使用雙向的(bidirectional)LSTM，而且如果沒有使用雙向的 LSTM 或者疊多層一點時，這個模型又很容易訓

練不起來，也就是說就會再次卡在 margin 設的值。但是 dot_1 的內積是兩個三維矩陣下去做的，下個段落會說明這的 dot_1 的用途。



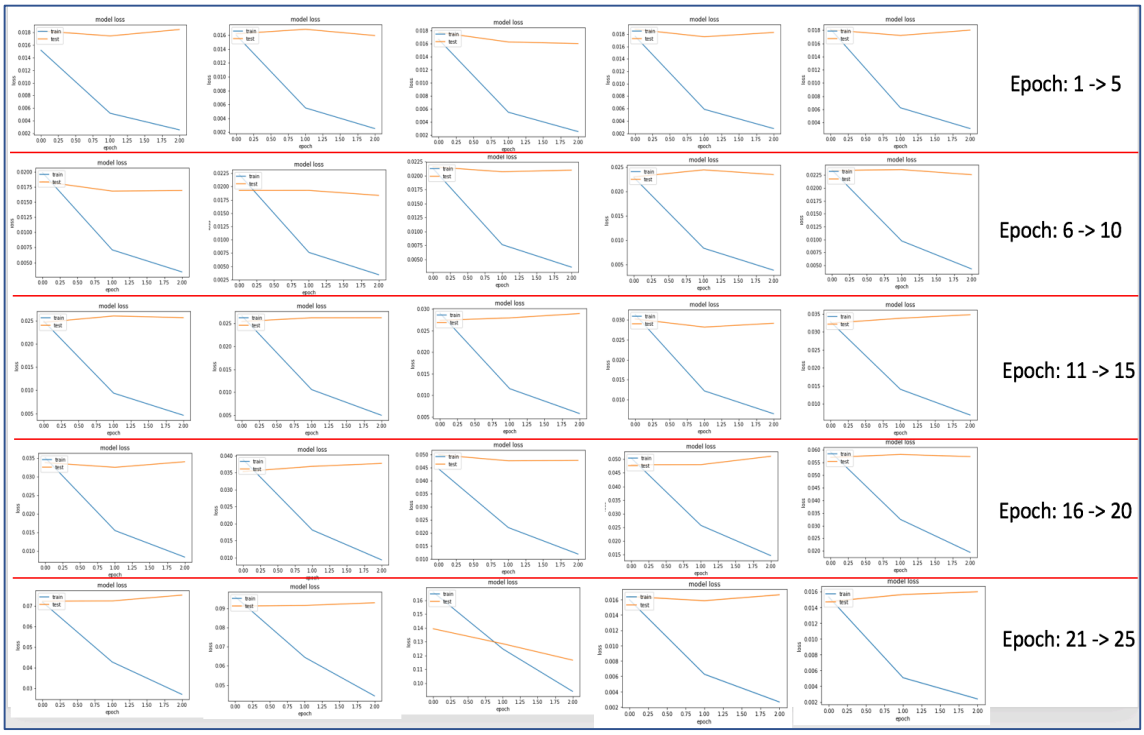
Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	(None, 246)	0	
input_1 (InputLayer)	(None, 246, 39)	0	
embedding_1 (Embedding)	(None, 246, 300)	3492900	input_2[0][0]
bidirectional_1 (Bidirectional)	(None, 246, 256)	172032	input_1[0][0]
bidirectional_2 (Bidirectional)	(None, 246, 256)	439296	embedding_1[0][0]
dropout_1 (Dropout)	(None, 246, 256)	0	bidirectional_1[0][0]
dropout_2 (Dropout)	(None, 246, 256)	0	bidirectional_2[0][0]
dot_1 (Dot)	(None, 256, 256)	0	dropout_1[0][0] dropout_2[0][0]
flatten_1 (Flatten)	(None, 62976)	0	dropout_1[0][0]
flatten_2 (Flatten)	(None, 65536)	0	dot_1[0][0]
dense_1 (Dense)	(None, 20)	1259540	flatten_1[0][0]
dense_2 (Dense)	(None, 20)	1310740	flatten_2[0][0]
dot_2 (Dot)	(None, 1)	0	dense_1[0][0] dense_2[0][0]

Total params: 6,674,508
Trainable params: 6,674,508
Non-trainable params: 0

在 embedding layer 的 weights，我仍然是使用一開始說明 word2vec 去做，但是我把出現次數小於三次的字數(min_count=3)都不下去訓練了，且 size=300,workers=5，詞向量的參數大概就是這樣，並且要跟著模型一起下去訓練(Trainable=True)。

以上的模型我稱為 qa_model，在訓練時候必須要建立另外兩個 contrasive_model 還有一個 prediction_model，這兩個模型在 compile 的時候必須要建立一個 dummy_variable(也就是一個零向量)，並且在 compile 的時候把自己 custom 的 loss 返回(cosine_similarity 的值)。而上面有一個地方有一個 Dot，我查到的資料是說那是一個 attention，用來在因為文章很長的情況下希望關注於在某特別重點的部分，因為我想說我的 input time stamp 是 246，而 caption 只有 15，所以我應該該是滿足上述的要求，當我放進去時上傳時 kaggle 分數都可以在 0.5 左右了（從 0.42->0.5）。而在 hinge_loss 的 margin 我選擇是 0.2，我沒有調整過這個的參數，我主要調整是在 LSTM 的個數、層數還有 dropout 的值。最後總結的參數是：使用 Bidirectional LSTM (192)，Dropout(0.3)，每一層後面都必須要接上 dropout。

而訓練因為要產生負樣本(negative sample)，所以我繪出 loss 的示意圖，會因為每一組正負樣本都會訓練三個 epoch 之後，必須要更換一組新的樣本組而形成二十五組 loss 圖表，以下是我整理過後過後的圖：

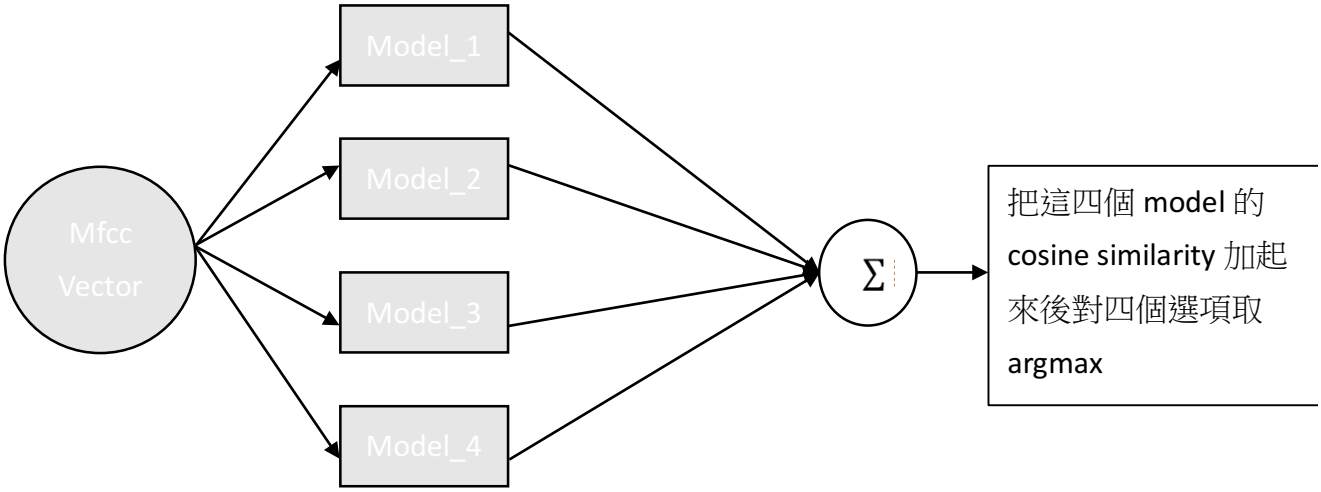


藍線：train

橘線：test

其實這個對於我來講算是比較好的一次成果，就是說在訓練集上面其實真的就是一直下降，但是在 validation 上面也是有下降，到最後時候可能是收斂了，validation 大概卡在 hinge loss -> 0.0147，而訓練集上面大概是 0.0039 甚至可以達到更低，最後我為了避免 overfitting 的情況，所以就在 25 epochs 的時候停止訓練了，上傳 kaggle 的分數為：0.54700，也是勉強過了 strong baseline。因為自己周遭同學討論的情形，用這個模型似乎是可以達到更高的分數，我覺得是我在 feature extraction(embedding)那邊的處理應該可以在修改了什麼，但是因為時間的關係我就沒去嘗試，最後使用 ensemble 看能不能衝至前 30%。

以下就來說最後實現 ensemble 的想法：



原本想說可以在像老師說：為了避免有些模型是比較不好的，可以訓練一個分類器(classifier)，可是這樣子對於我的訓練來講感覺是不適用的，因為我的 task 是 unsupervised 的，好像有點沒辦法實作出來。但是這樣直接相加的對於我的成果反而也是挺不錯的，我的想法是說：在我取正負樣本的時候可能有些模型有學到另一些模型沒看過負樣本(negative sample)的值，所以在把它相加的情況下，有些模型可能在有兩個選項的 cosine similarity 相近的情況下，就可以在這個情況做出一個較正確的答案。而這樣上傳 kaggle 的分數是 0.584、0.59910，效果真的是非常顯著的。

4. 總結：

這個期末專題的比較難的地方應該是在於說他的訓練過程需要非常的細心去調整參數，以及增加自己的模型容量(model capacity)。舉例來說：我最後這個模型在我沒有加進去 dropout 時，好像是受到 LSTM 的 gradient explosion 的影響，會直接卡在一個 hinge loss　margin 設的值，當我一加進去 dropout 時，整個模型都跑起來了，之前卡著很久在第一階段 strong baseline 沒有過應該是出在這個小錯誤。最後就是 ensemble 的處理方法若不是都是我自己訓練出來的模型可能就沒辦法直接相加了，沒有針對於較好的模型給予較大的參數(weights)。至於最後比較遺憾的就是沒有去很了解 attention 機制，沒有使用 keras 寫

出來是滿可惜的，因為一開始長時間都投資在 seq2seq 的模型。最後還是要謝謝助教的耐心回答我的所有問題！